

LU0a - Arrays - TBD

Einleitung

In der Programmierung stösst man schnell auf Situationen, in denen man nicht nur einen einzelnen Wert, sondern eine **Sammlung von Werten speichern** möchte – zum Beispiel eine Liste mit Namen, Zahlen oder Objekten.

In JavaScript gibt es dafür den Datentyp Array. Ein Array ist eine geordnete Liste, in der jedes Element über einen Index angesprochen werden kann (beginnend bei 0).

Arrays sind unglaublich vielseitig: Man kann Elemente hinzufügen, entfernen, sortieren oder mit speziellen Methoden wie `map` und `filter` weiterverarbeiten.

Was ist ein Array?

Ein Array ist eine geordnete Sammlung von Werten. Diese Werte können Zahlen, Strings, Objekte oder sogar andere Arrays sein. Arrays sind besonders praktisch, wenn man mehrere Werte unter einem Namen speichern möchte.

Beispiel:

```
let zahlen = [1, 2, 3, 4, 5];  
let namen = ["Anna", "Ben", "Clara"];
```

Zugriff auf Array-Elemente

Jedes Element in einem Array hat einen Index, beginnend bei 0.

```
let farben = ["rot", "gruen", "blau"];  
  
console.log(farben[0]); // "rot"  
console.log(farben[1]); // "gruen"  
console.log(farben[2]); // "blau"
```

Array-Laenge

Mit der Eigenschaft `.length` erfährst du, wie viele Elemente im Array sind:

```
let zahlen = [10, 20, 30];  
console.log(zahlen.length); // 3
```

Elemente hinzufügen oder entfernen

JavaScript stellt praktische Methoden bereit:

```
let tiere = ["Hund", "Katze"];

// Hinzufügen
tiere.push("Maus");      // ans Ende
tiere.unshift("Vogel");  // an den Anfang
console.log(tiere);      // ["Vogel", "Hund", "Katze", "Maus"]

// Entfernen
tiere.pop();             // entfernt letztes Element
tiere.shift();           // entfernt erstes Element
console.log(tiere);      // ["Hund", "Katze"]
```

Iteration über Arrays

Mit einer Schleife kannst du über alle Elemente gehen:

```
let zahlen = [1, 2, 3];

for (let i = 0; i < zahlen.length; i++) {
  console.log(zahlen[i]);
}

// moderne Variante:
zahlen.forEach((wert) => console.log(wert));
```

Besondere Methoden

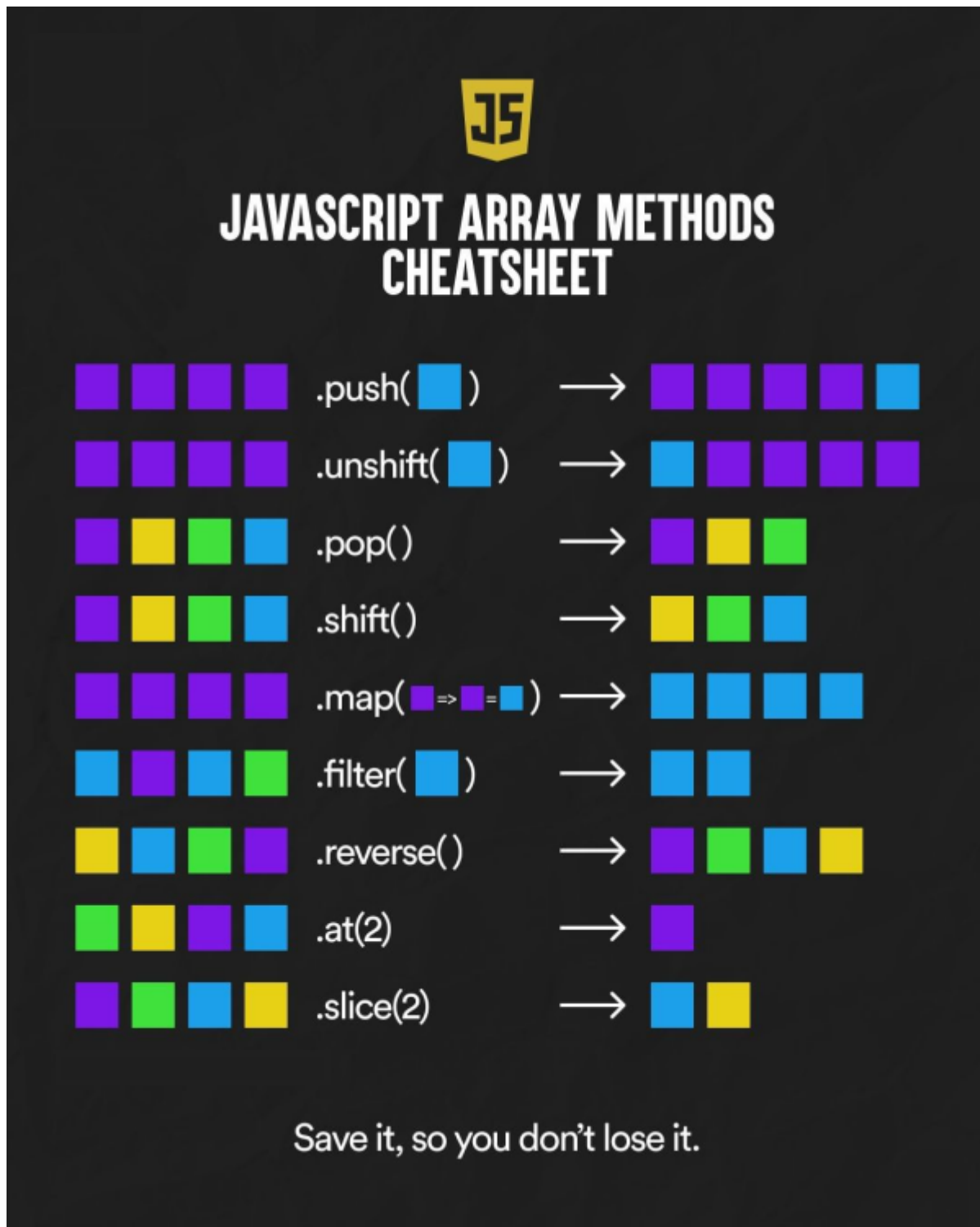
- map → erstellt ein neues Array basierend auf jedem Element.
- filter → filtert Elemente nach Bedingung.
- find → findet das erste passende Element.

```
let zahlen = [1, 2, 3, 4, 5];
let verdoppelt = zahlen.map(x => x * 2);
console.log(verdoppelt); // [2, 4, 6, 8, 10]

let gerade = zahlen.filter(x => x % 2 === 0);
console.log(gerade); // [2, 4]

let ersteGrosse = zahlen.find(x => x > 3);
console.log(ersteGrosse); // 4
```

Übersicht über Array-Methoden



The cheatsheet features a yellow JavaScript logo at the top center. Below it, the title "JAVASCRIPT ARRAY METHODS CHEATSHEET" is written in large, bold, white capital letters. The methods are listed in a grid-like format, each with a visual representation of the array state before and after the operation. The methods shown are: .push(), .unshift(), .pop(), .shift(), .map(), .filter(), .reverse(), .at(), and .slice(). Each row shows the initial array of colored squares, the method call, an arrow, and the resulting array. At the bottom, the text "Save it, so you don't lose it." is displayed in white.

JAVASCRIPT ARRAY METHODS CHEATSHEET

Initial Array	Method	Resulting Array
[Purple, Purple, Purple, Purple]	.push(Blue)	[Purple, Purple, Purple, Purple, Blue]
[Purple, Purple, Purple, Purple]	.unshift(Blue)	[Blue, Purple, Purple, Purple, Purple]
[Purple, Yellow, Green, Blue]	.pop()	[Purple, Yellow, Green]
[Purple, Yellow, Green, Blue]	.shift()	[Yellow, Green, Blue]
[Purple, Purple, Purple, Purple]	.map(Purple => Blue)	[Blue, Blue, Blue, Blue]
[Blue, Purple, Blue, Green]	.filter(Blue)	[Blue, Blue]
[Yellow, Blue, Green, Purple]	.reverse()	[Purple, Green, Blue, Yellow]
[Green, Yellow, Purple, Blue]	.at(2)	[Purple]
[Purple, Green, Blue, Yellow]	.slice(2)	[Blue, Yellow]

Save it, so you don't lose it.

Youtube-Lernvideo

[arrays_javascript_tutorial_deutsch_teil_12.mp4](#)

Zusatzmaterial

- [W3School- JavaScript Arrays](#)
- [SelfHTML - JS Arrays](#)



Volkan Demir

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/de/modul/m288/learningunits/lu03/01?rev=1756750921>

Last update: **2025/09/01 20:22**

